## Passo 1: Analisar os Requisitos

Os requisitos foram fornecidos na forma de Histórias de Usuário com Critérios de Aceitação.

## Passo 2: Escrever Cenários de Teste

### Para a História de Usuário 1:

**Cenário:** Adicionar uma tarefa apenas com um título

**Cenário:** Adicionar uma tarefa com um título e descrição detalhada

**Cenário:** Adicionar uma tarefa com prazo

**Cenário:** Adicionar uma tarefa sem título (esperar falha)

**Cenário:** Verificar se a tarefa recém-adicionada está na lista de tarefas pendentes

### Para a História de Usuário 2:

**Cenário:** Marcar uma tarefa como concluída

**Cenário:** Visualizar tarefas concluídas na seção de concluídas

**Cenário:** Desmarcar uma tarefa concluída

**Cenário:** O horário de conclusão da tarefa é registrado

**Cenário:** O status da tarefa é refletido com precisão na interface do usuário

## Passo 3: Implementar Cenários de Teste com Cucumber

Você escreverá esses cenários em um arquivo .feature. Aqui está um exemplo de como escrever os cenários na sintaxe Gherkin para a primeira História de Usuário:

### Recurso: Gerenciamento de Tarefas

**Cenário:** Adicionar uma tarefa apenas com um título

Dado que estou logado no sistema de gerenciamento de tarefas

Quando escolho adicionar uma nova tarefa

E eu digito "Comprar leite" como o título

E eu envio a tarefa

Então devo ver "Comprar leite" na lista de tarefas pendentes

**Cenário:** Adicionar uma tarefa sem título

Dado que estou logado no sistema de gerenciamento de tarefas

Quando escolho adicionar uma nova tarefa

E eu envio a tarefa sem título

Então devo ver uma mensagem de erro

Mais cenários para o restante dos critérios de aceitação...

## Passo 4: Escrever Definições de Passos para o Cucumber

Para cada passo no seu arquivo .feature, você precisará escrever uma definição de passo correspondente. Essas são escritas na linguagem de sua escolha suportada pelo Cucumber (por exemplo, Ruby, Java, JavaScript). Aqui está um exemplo em Ruby:

Dado('que estou logado no sistema de gerenciamento de tarefas') do

Código para fazer login no sistema

end

Quando('escolho adicionar uma nova tarefa') do

Código para navegar até o formulário de adição de tarefa

end

E('eu digito {string} como o título') do |title|

Código para digitar o título no formulário

end

E('eu envio a tarefa') do

Código para enviar o formulário

end

Então('devo ver {string} na lista de tarefas pendentes') do |title|

Código para verificar se a tarefa está na lista de tarefas pendentes

end

Então('devo ver uma mensagem de erro') do

Código para verificar se uma mensagem de erro é exibida

end]

Definições de passos adicionais para outros cenários...

## Passo 5: Implementação de Código

Este passo envolve configurar um projeto na sua linguagem escolhida e escrever o código para fazer os testes passarem.

Exemplo de Configuração de Projeto para Ruby com Cucumber e Capybara:

Crie um Gemfile com as gemas necessárias:

source 'https://rubygems.org'

gem 'cucumber'

gem 'capybara'

gem 'selenium-webdriver'

Instale as Gemas:

bundle install

Configure o Cucumber:

cucumber --init

Escreva o código da aplicação com o qual as definições de passos irão interagir. Por exemplo, se você estiver criando uma aplicação web, você configuraria rotas, visões e modelos para as tarefas.

Execute os testes do Cucumber para vê-los falhar (fase Vermelha do TDD).

Implemente as funcionalidades necessárias para passar nos testes (fase Verde do TDD).

Refatore seu código conforme necessário.

Repita o ciclo de vermelho-verde-refatoração até que todos os testes passem e a aplicação funcione como esperado.

## Passo 6: Escrever Testes de Aceitação

Por fim, você escreveria testes de aceitação para os critérios de aceitação fornecidos. Estes são testes de alto nível que garantem que o sistema funcione do ponto de vista do usuário.

Exemplo de Teste de Aceitação:

recurso 'Gerenciamento de Tarefas' do

cenário 'Usuário adiciona uma nova tarefa' do

visit new\_task\_path

preencher 'Título', com: 'Comprar leite'

clicar no botão 'Criar Tarefa'

expect(page).to have\_content('Tarefa foi criada com sucesso.')

expect(page).to have\_content('Comprar leite')

end

end"